EMU Working Group
Internet Draft
Intended status: Informational

P. Urien
Telecom ParisTech
C.Kiennert
Telecom ParisTech

October 15, 2009

Expires: April 15, 2010

EAP BIO
draft-urien-kiennert-emu-bio-00.txt

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 2010.

Abstract

EAP-TTLS is an EAP method that provides secured authentication as described in RFC 5281. This method makes generally use of two phases in order to complete authentication. The first one consists in the authentication of the TTLS server to the client, established by a TLS handshake between the client and the TTLS server. The handshake may be either mutual or one-way. The authentication of the client to the server may then be negotiated during phase two of EAP-TTLS, thanks to widely-deployed authentication mechanisms such as CHAP, PAP, MS-CHAP or MS-CHAP-V2.

The purpose of EAP-BIO is to define how to use a biometric authentication mechanism during phase two of EAP-TTLS. This authentication mechanism ranges from physiological characteristics such as fingerprint identification, to behavioral characteristics such as voice or signature analysis. Hence, EAP-BIO combines the security features of EAP-TTLS and biometric authentication.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

Table of Contents

1 Overview

The Extensible Authentication Protocol (EAP) [RFC 3748] defines a standard message exchange that allows a server to authenticate a client using an authentication method agreed upon by both parties. EAP may be extended with additional authentication methods by registering such methods with IANA or by defining vendor specific methods.

EAP Tunneled TLS (EAP-TTLS) [RFC 5281] defines an authentication protocol performing mutual authentication between the client and the server. This authentication can be directly negotiated during the TLS handshake if the client uses a certificate, but contrary to EAP-TLS such a configuration is not required from the client. If the client cannot authenticate to the server with a certificate, then the TLS handshake is used as a first phase allowing the client and the server to implicitly generate keying material in order to create a secure tunnel where the following data are to be transferred. The second phase of EAP-TTLS allows the client to securely authenticate to the server using common protocols such as CHAP, PAP, MS-CHAP or MS-CHAP-V2. Nevertheless, any authentication protocol, such as biometric mechanisms, may be defined.

Biometric authentication involves the use of physical and/or behavioural characteristics of individuals to identify them and to control access to places or things, such as computerized equipment, or more specifically, applications running on that equipment. Biometrics has some advantages over more conventional authentication techniques (login and password, PIN codes, etc.) since nothing has to be carried or remembered that might be stolen. Among the many biometric technologies in use are fingerprint analysis, hand geometry analysis, retina scanning, iris scanning, signature analysis, facial recognition, keystroke analysis, and voice analysis.

Based on original measurement of a biometric characteristic (i.e. enrolment), a person's identity can thereafter be verified automatically when requesting access to a computer application or to any other resource by re-sampling the characteristic and comparing the biometric data with the enrolment. If a sufficiently close match is found, then the identity is verified.

However, the time needed for biometric comparisons with samples in database is much longer than the time required for more widely-deployed authentication mechanisms. Thus, the protocol MUST ask a login from the user before processing its biometric characteristic. Such an implementation will significantly reduce the time required for the authentication since the server only needs to compare the biometric characteristic with one sample from the database.

2 Terminology

AAA

Authentication, Authorization and Accounting - functions that are generally required to control access to a network and support billing and auditing.

AAA protocol

A network protocol used to communicate with AAA servers; examples include RADIUS and Diameter.

AAA server

A server which performs one or more AAA functions: authenticating a user prior to granting network service, providing authorization (policy) information governing the type of network service the user is to be granted, and accumulating accounting information about actual usage.

AAA/H

A AAA server in the user's home domain, where authentication and authorization for that user are administered.

Access Point

A network device providing users with a point of entry into the network, and which may enforce access control and policy based on information returned by a AAA server. Since the access point terminates the server side of the EAP conversation, for the purposes of this document it is therefore equivalent to the "authenticator" as used in the EAP specification [RFC 3748].
Since the access point acts as a client to a AAA server, for the purposes of this document it is therefore also equivalent to the "NAS", as used in AAA specifications such as [RFC 2865].

Access Domain

The domain, including access points and other devices, that provides users with an initial point of entry into the network; for example, a wireless hot spot.

Client

A host or device that connects to a network through an access point; since it terminates the client side of the EAP conversation, for the purposes of this document, it is therefore equivalent to the "peer", as used in the EAP specification [RFC 3748].

Domain

A network and associated devices that are under the administrative control of an entity such as a service provider or the user's home organization.

Minutia

A generic term in biometrics that designates specific points in a fingerprint record such as ridge bifurcations or endpoints.

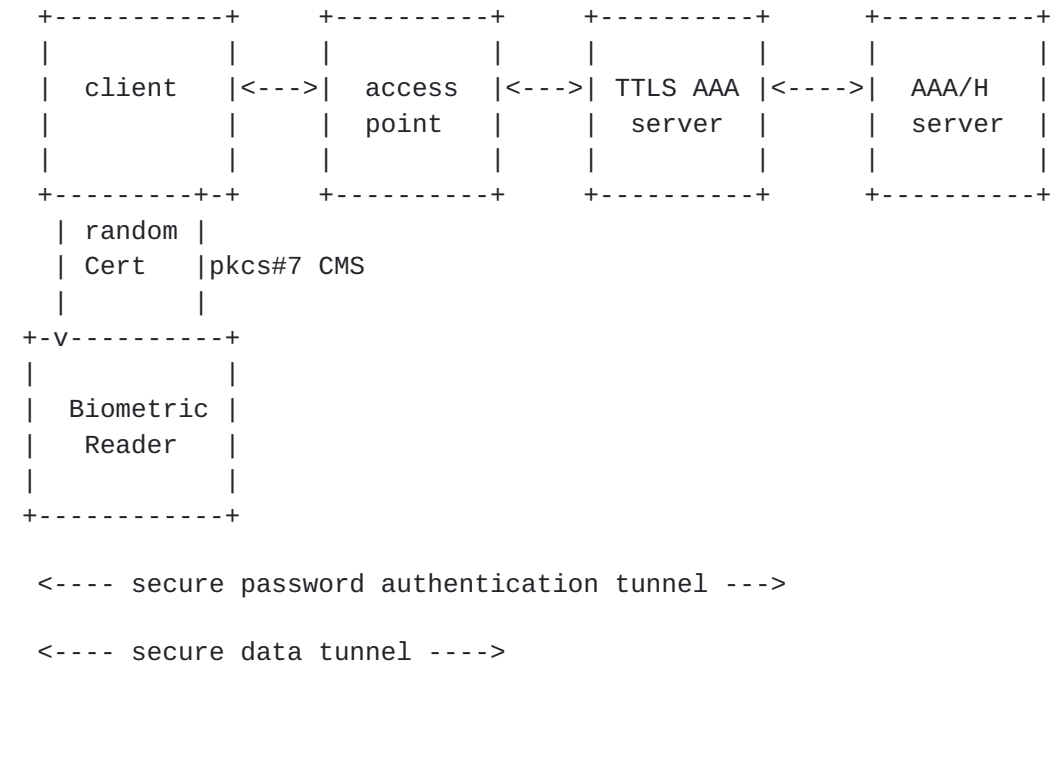
TTLS server

A AAA server which implements EAP-TTLS. This server may also be capable of performing user authentication, or it may proxy the user authentication to a AAA/H.

User

The person operating the client device; though the line is often blurred, "user" is intended to refer to the human being who is possessed of an identity (username), password or other authenticating information, and "client" is intended to refer to the device which makes use of this information to negotiate network access. There may also be clients with no human operators; in this case the term "user" is a convenient abstraction.

3 Functional Architecture



The architecture depicted above only displays the general organization for message transmission, which means that additional security features are not visible. An overview of the security offered by this architecture will be described below.

3.1 Carrier Protocols

As explained in RFC 5281, the entities shown above, except the biometric reader and the client, communicate with each other using carrier protocols capable of encapsulating EAP. The client and the access point communicate typically using a link layer carrier protocol such as PPP or EAPOL. The access point, TTLS server and AAA/H server communicate using a AAA carrier protocol such as RADIUS or Diameter.

EAP, and therefore EAP-TTLS, must be initiated via the carrier protocol between the client and the access point. In PPP or EAPOL, for example, EAP is initiated when the access point sends an EAP-Request/Identity packet to the client.

The keying material used to encrypt and authenticate the data connection between the client and the access point is developed implicitly between the client and TTLS server as a result of EAP-TTLS negotiation. This keying material must be communicated to the access point by the TTLS server using the AAA carrier protocol.

3.2 Security overview

The architecture inherits all the security features of EAP-TTLS, and as such allows secure authentication between the client and the TTLS server. However, the biometric elements have to be implemented with proper security in order not to compromise the whole new authentication protocol.

Some of the potential attacks related to biometrics may indeed compromise the security of the protocol and may result in unauthorized access:
- If the user gives a false sample to the biometric reader (false finger, mask, eye picture), the reader may grant access according to the false credentials it was given.

- If the user manages to send previously acquired samples from another user, the protocol may grant access if it has no protection against replay attacks.

These are two potential attacks that are not covered by the security offered by EAP-TTLS. The security features addressing these flaws mainly rely on smart cards and digital signatures, and will be discussed in later sections.

4 EAP-BIO Overview

Since EAP-BIO relies on EAP-TTLS, the authentication process is the same as EAP-TTLS protocol. It consists of two phases, where phase 1 is a TLS handshake that either performs mutual authentication, or authenticates the TTLS server to the client and allows a secure tunnel to be created, in which authentication data and other significant information will be exchanged during phase 2 of the protocol.

4.1 Phase 1: Handshake

In phase 1, the TLS handshake protocol is used to authenticate the TTLS server to the client and, optionally, to authenticate the client to the TTLS server.

The TTLS server initiates the EAP-TTLS method with an EAP-TTLS/Start packet, which is an EAP-Request with Type = EAP-TTLS and the S (Start) bit set. This indicates to the client that it should begin TLS handshake by sending a ClientHello message.

EAP packets continue to be exchanged between client and TTLS server to complete the TLS handshake, as described in [RFC 5216]. Phase 1 is completed when the client and the TTLS server exchange ChangeCipherSpec and Finished messages. At this point, additional information may be securely tunnelled.

As part of the TLS handshake protocol, the TTLS server will send its certificate along with a chain of certificates leading to the certificate of a trusted CA. The client will need to be configured with the certificate of the trusted CA in order to perform the authentication.

If certificate-based authentication of the client is desired, the client must have been issued a certificate and must have the private key associated with that certificate.

4.2 Phase 2: Tunneled Authentication

In phase 2, the TLS Record Layer is used to securely tunnel information between the client and the TTLS server. This information is encapsulated in sequences of attribute-value pairs (AVPs), whose use and format are described in the next paragraph.

The server will ask the client to send its credentials for authentication, and before asking for biometric data, the server MUST ask the user's login. The user will then be asked to submit his biometric characteristic to the reader.

Such a procedure will allow the biometric characteristic to be compared only with the sample of the specified user, instead of testing all the database samples. Since the comparison algorithm may be complex, a significant amount of time may be saved for the authentication.

The login and the biometric characteristic are encapsulated in AVPs, and then transmitted to the server through the secured TLS tunnel. In accordance with the EAP-TTLS protocol, other data may be exchanged between the client and the server using AVPs.

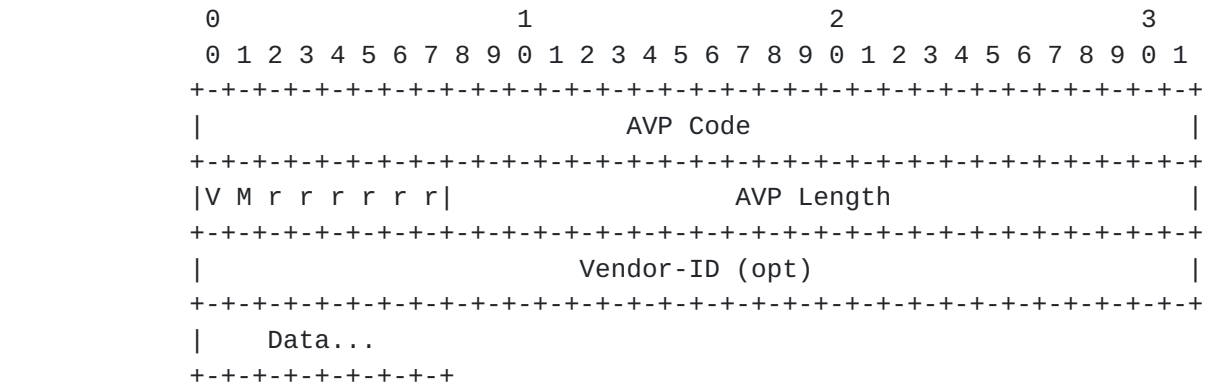
The TTLS server recovers the AVPs in clear text from the TLS record layer. If the AVP sequence includes authentication information, it forwards this information to the AAA/H server using the AAA carrier protocol. Note that the EAP-TTLS and AAA/H servers may be one and the same, in which case it simply processes the information locally.

When the TTLS server has gathered enough information, it issues either an EAP-Success or EAP-Failure. Thus, if the AAA/H rejects the client based on forwarded authentication information, the TTLS server MUST issue an EAP-Failure. If the AAA/H accepts the client, the TTLS server MUST issue an EAP-Success.

The TTLS server distributes data connection keying information and other authorization information to the access point in the same AAA carrier protocol message that carries the EAP-Success.

4.3 AVP format

The format of an AVP is shown below. All items are in network, or big-endian, order; that is, they have most significant octet first.



AVP Code

The AVP Code is four octets and, combined with the Vendor-ID field if present, identifies the attribute uniquely. The first 256 AVP numbers represent attributes defined in RADIUS. AVP numbers 256 and

above are defined in Diameter. The codes used in EAP-BIO are yet to be defined.

AVP Flags

The AVP Flags field is one octet, and provides the receiver with information necessary to interpret the AVP.

The 'V' (Vendor-Specific) bit indicates whether the optional Vendor-ID field is present. When set to 1, the Vendor-ID field is present and the AVP Code is interpreted according to the namespace defined by the vendor indicated in the Vendor-ID field.

The 'M' (Mandatory) bit indicates whether support of the AVP is required. If this bit is set to 0, this indicates that the AVP may be safely ignored if the receiving party does not understand or support it. If set to 1, this indicates that the receiving party must fail the negotiation if it does not understand the AVP; for a TTLS server, this would imply returning EAP-Failure, for a client, this would imply abandoning the negotiation.

The 'r' (reserved) bits are unused and must be set to 0.

AVP Length

The AVP Length field is three octets, and indicates the length of this AVP including the AVP Code, AVP Length, AVP Flags, Vendor-ID (if present) and Data.

Vendor-ID

The Vendor-ID field is present if the 'V' bit is set in the AVP Flags field. It is four octets, and contains the vendor's IANA-assigned "SMI Network Management Private Enterprise Codes" [9] value. Vendors defining their own AVPs must maintain a consistent namespace for use of those AVPs within RADIUS, Diameter and EAP-TTLS.
A Vendor-ID value of zero is equivalent to absence of the Vendor-ID field altogether.

4.4 AVP Sequences

Data encapsulated within the TLS Record Layer must consist entirely of a sequence of zero or more AVPs. Each AVP must begin on a 4-octet boundary relative to the first AVP in the sequence. If an AVP is not a multiple of 4 octets, it must be padded with 0s to the next 4-octet boundary.

Note that the AVP Length does not include the padding.

5 Signatures and Smart Cards

Though EAP-TTLS offers high security for authentication, the biometric equipments might be flawed and compromise the whole EAP-BIO protocol. The potential security flaws described in section 3.2 MUST be addressed with smart cards and cryptographic solutions such as digital signatures. The number of signatures used in the protocol will depend on the level of security required.

5.1 Elements to be secured

The most important element to be secured should be the biometric reader. In order to prevent material flaws, the reader MUST be certified as secure. If it is not, a signature from the reader would have no meaning. The secure biometric reader should then be affected a smart card in order to store the private key required for the signature. Thus, the biometric characteristic SHOULD be signed by the reader, along with a trusted timestamp in order to prevent replay attacks.

The client, whose the biometric characteristic is transmitted to, SHOULD also sign the characteristic before sending it to the TTLS server through the TLS record layer.

The third element which may require specific security is the user. Depending on the use case of the EAP-BIO protocol, the authentication may require a smart card from the user, containing his personal public-key certificate. Such a security level may be required in places such as airports.

Hence, the biometric authentication requires two or three signatures, depending of the security requested.

5.2 PKCS #1 Signature

The signature of the biometric characteristic, along with a trusted timestamp in the case of the biometric reader, is computed according to the PKCS #1 specifications [RFC 3447]. It is here encapsulated in the data field of EAP-TTLS phase 2 AVPs (fourth octet and following).

The signature is generated from two steps. The first one consists in encoding the message using EMSA-PSS, described in section 9.1.1 of RFC 3447. Since this is a probabilistic function, the signer has to choose not only a hash function, but also a mask generation function and a salt. The second one consists in the output of the RSA signature. This signature is calculated out of three steps:

a. Converting the hashed message to an integer message representative.

b. Applying a RSA signature primitive (described in section 5.2.1 of RFC 3447) to the private key and the message representative to produce an integer signature representative.

c. Converting the signature representative to a signature of k octets, where k is the length in octets of the RSA modulus n.

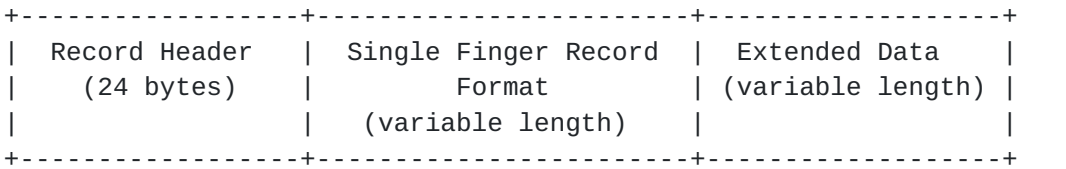
6 Biometric Formats

Many biometric formats, protocols and definitions have been standardized by international organizations such as ISO (International Organization for Standardization), IEC (International Electrotechnical Commission) or JTC1 (Joint Technical Committee One).

EAP-BIO shall use the CBEFF (Common Biometric Exchange Formats Framework) data structure, a standard regarding biometrics which describes the necessary data to be inserted in the headers so that different systems may exchange biometric data within only one file. The CBEFF data structure is composed of three parts: the Standard Biometric Header, the Biometric Data Block, and the Signature Block. This data structure is used as a reference in the standards defined for the different biometric characteristics.

6.1 Fingerprint format

The requirements for fingerprint analysis are described in the [ISO/IEC 19794] standard. It specifies the content, format and measure units from a fingerprint picture to be used in identification or authentication of a user.
The basic entities for fingerprint analysis are ridges and valleys of the finger pattern, and more particularly specific points such as ridge bifurcations (where a ridge is divided in two) and ridge endpoints (where a ridge suddenly stops). The layout of these points is specific to each individual, and as such represents significant information for identification or authentication.
The format of a Finger Minutiae Record is described in the ISO/IEC 19794-2 standard, and can be represented as follows:



The header contains information about the characteristics and options of the recording process (equipment, image size, resolution, number of views).

The Single Finger Record Format block is divided in two parts: the Finger Header block (4 bytes) and the Finger Minutiae Data blocks (6 bytes per block). The Finger Header contains information about the finger position, its quality, as well as the number of minutiae, i.e. bifurcations or endpoints recorded on the fingerprint. The Finger Minutiae Data blocks contain technical data about each minutia recorded (position, angle, quality). One block is added for each minutia.

The Extended Data fields may contain any relevant information about the finger minutiae or about the record that could be of use to the equipment retrieving the fingerprint data. The length of these fields (there may be several of them) should be as small as possible.

6.2 Fingerprint Card format

The ISO/IEC 19794 standard also defines two possible formats for cards: the Normal Size Finger Minutiae Format, in which the minutiae data are encoded on five octets for each minutia (containing the type, coordinates and angle), and the Compact Size Finger Minutiae Format, in which these data are encoded with lower resolution on three octets.

7 PKCS #7 capsule

The biometric characteristic will be signed by two or three signers before being sent to the TTLS server through AVPs. The format described in [RFC 2315] defines two main types for the signature: SignerInfo and SignedData.

For each signer, the encrypted message digest and other signer-specific information are collected into a SignerInfo value. Certificates and certificate-revocation lists for each signer, and those not corresponding to any signer, are collected in this step. The defined format is the following:

```
SignerInfo ::= SEQUENCE {
version Version,
issuerAndSerialNumber      IssuerAndSerialNumber,
digestAlgorithm            DigestAlgorithmIdentifier,
authenticatedAttributes    [0] IMPLICIT Attributes OPTIONAL,
digestEncryptionAlgorithm  DigestEncryptionAlgorithmIdentifier,
encryptedDigest            EncryptedDigest,
unauthenticatedAttributes  [1] IMPLICIT Attributes OPTIONAL
}
EncryptedDigest ::= OCTET STRING
```

The timestamp required for the biometric reader signature may be inserted in the authenticatedAttributes field.

The message-digest algorithms and the SignerInfo values for all the signers are collected together with the content into a PKCS#7 SignedData structure (Object Identifier 1.2.840.113549.1.7.2), the format of which is defined as follows:

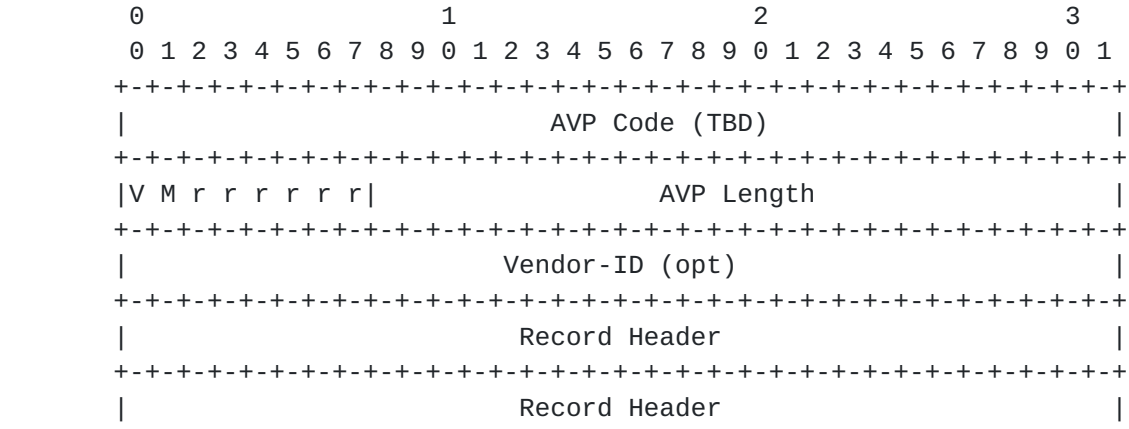
```
SignedData ::= SEQUENCE {
version Version,
digestAlgorithms DigestAlgorithmIdentifiers,
contentInfo ContentInfo,
certificates [0] IMPLICIT ExtendedCertificatesAndCertificates
OPTIONAL,
crls[1] IMPLICIT CertificateRevocationLists OPTIONAL,
signerInfos SignerInfos
}

DigestAlgorithmIdentifiers ::= SET OF DigestAlgorithmIdentifier

SignerInfos ::= SET OF SignerInfo

ContentInfo ::= SEQUENCE {
  contentType ContentType,
  content
    [0] EXPLICIT ANY DEFINED BY contentType OPTIONAL }

ContentType ::= OBJECT IDENTIFIER
```

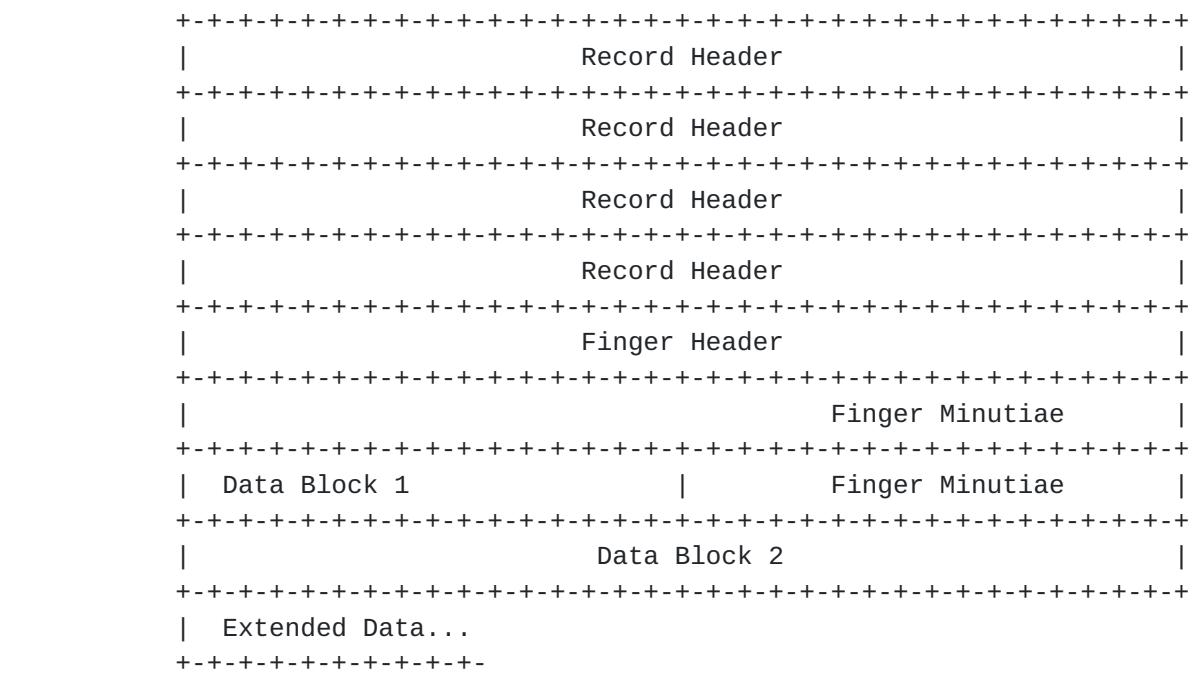
8 AVP examples

The EAP-BIO protocol makes use of AVPs in the second phase of the EAP-TTLS protocol. These AVPs contain information as described in the preceding sections, and can be represented as follows:

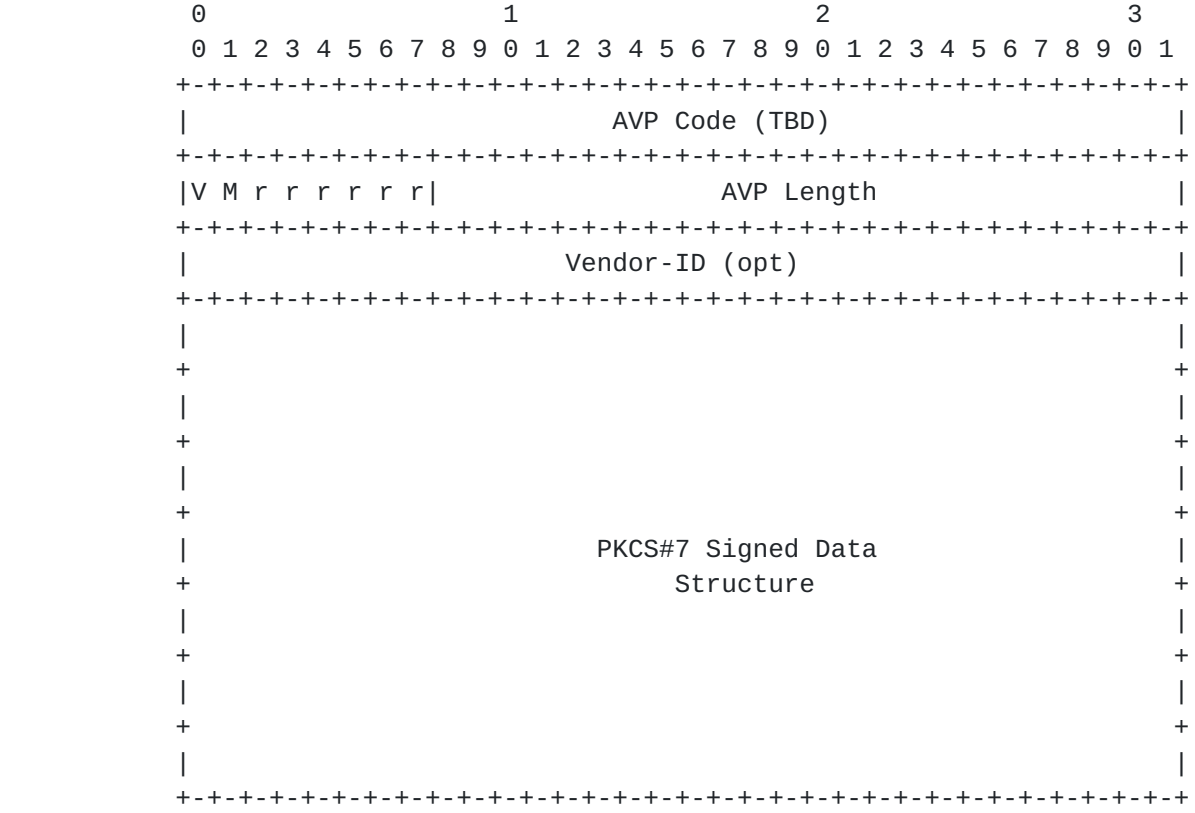
8.1 AVP containing biometric data

This example displays an AVP containing CBEFF fingerprint data comprised of two minutiae block of the same finger:

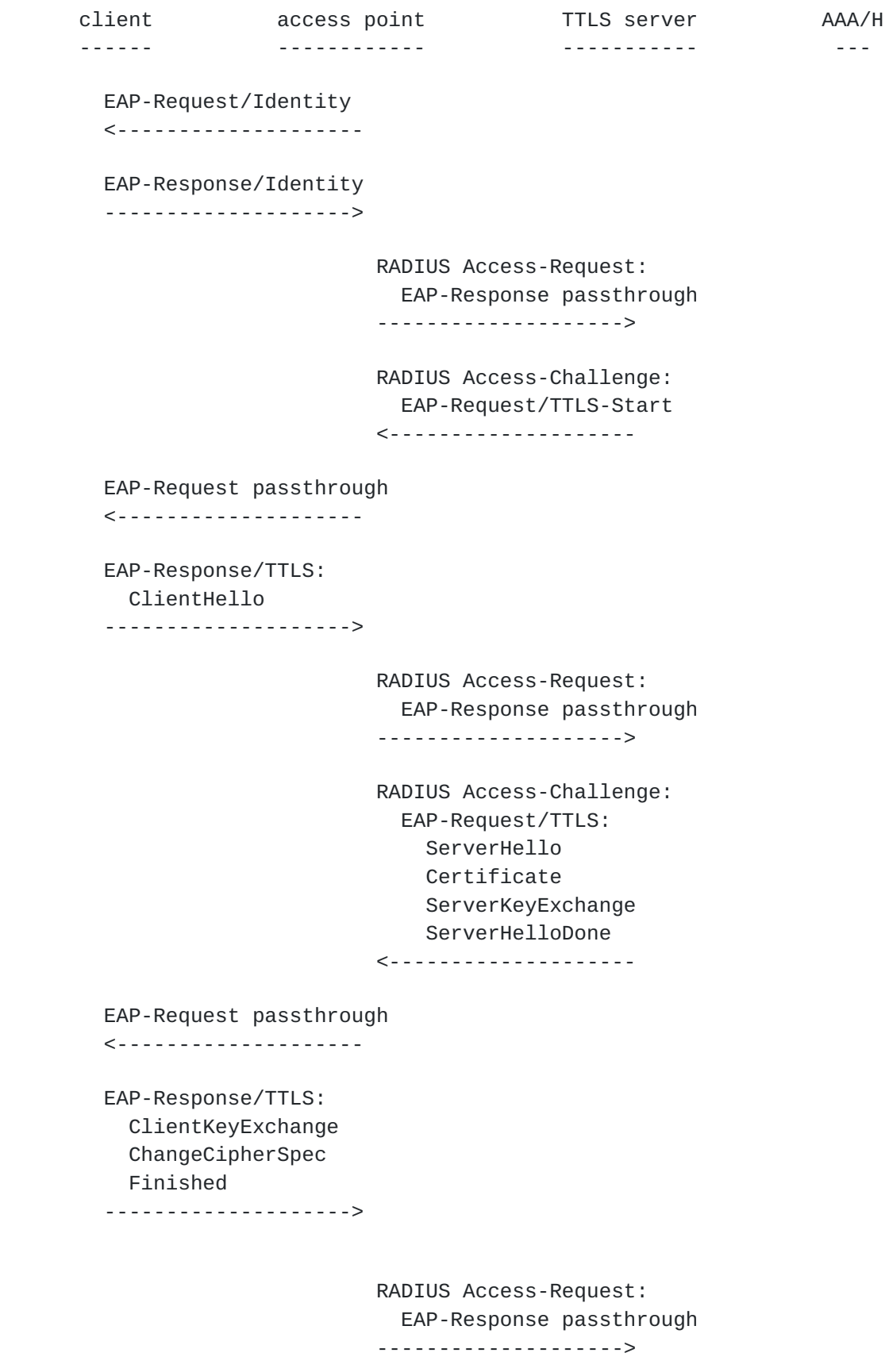


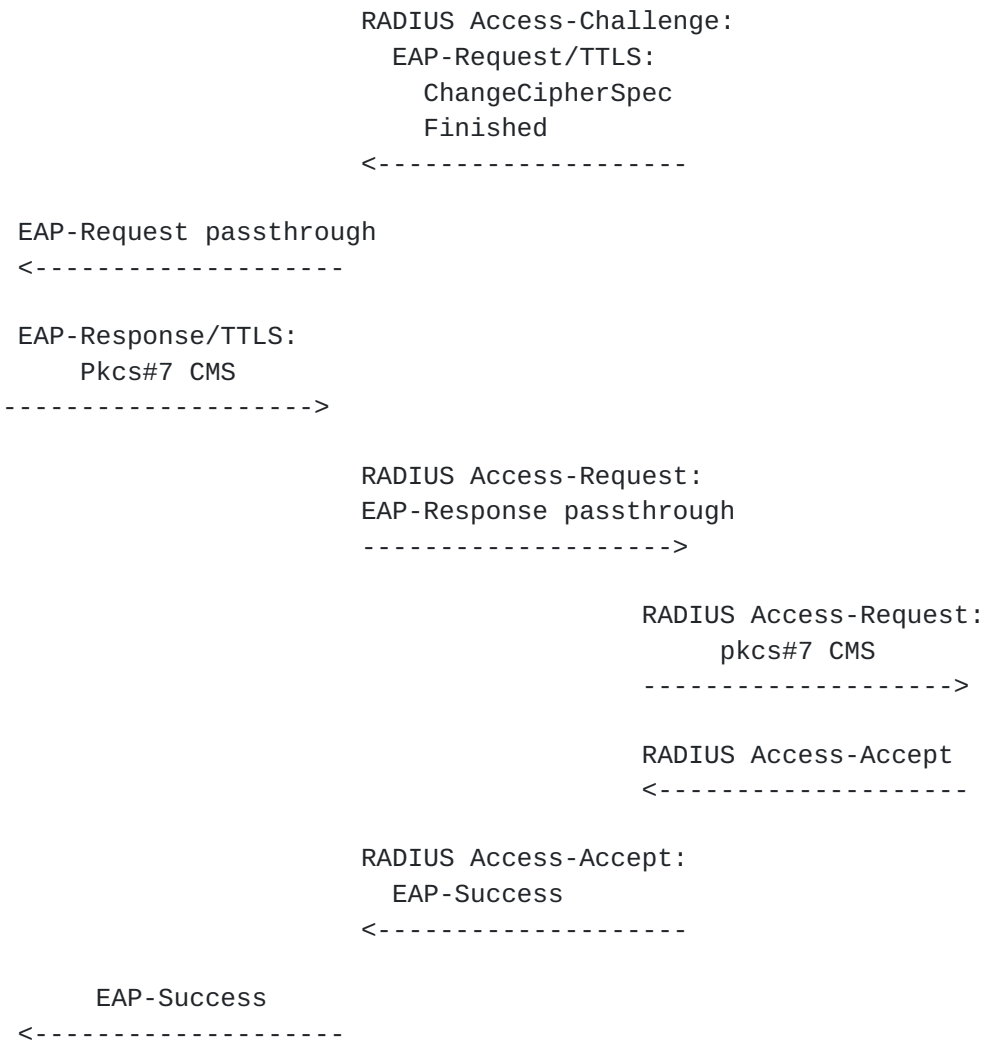


8.2 AVP containing a PKCS #7 capsule



9 Protocol details





10 IANA Considerations

11 Security Considerations

12 Normative References

[RFC 5281] Paul Funk, Simon Blake-Wilson, EAP Tunneled TLS Authentication Protocol Version 0 (EAP-TTLSv0), August 2008

[RFC 5216] D. Simon, B. Aboba, R. Hurst, "The EAP-TLS Authentication Protocol", March 2008.

[RFC 3748] B. Aboba, L. Blunk, J. Vollbrecht, J. C. Sun, H. Levkowetz, "Extensible Authentication Protocol (EAP)" RFC 3748, June 2004

[RFC 3447] J. Jonsson, B. Kaliski, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", February 2003

[RFC 2315] B. Kaliski, "PKCS #7: Cryptographic Message Syntax, Version 1.5", March 1998.

[RFC 2865] C. Rigney, S. Willens, A. Rubens, W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", June 2000.

[ISO/IEC 19794] BS ISO/IEC 19794 series. "Information technology. Biometric data interchange formats".

13 Non Normative References

14 Authors' Addresses

Pascal Urien
Telecom ParisTech
37/39 rue Dareau
75014 Paris Phone: NA
France Email: Pascal.Urien@enst.fr

Christophe Kiennert
Telecom ParisTech
37/39 rue Dareau
75014 Paris Phone: NA
France Email: Christophe.Kiennert@enst.fr

Full Copyright Statement